CLAIMS:

What is claimed is:

1. A method for buffering data in a device configured to be coupled to other devices in a loop network topology, the method comprising:

storing a plurality of successive words into storage locations in a circular memory;

reading the plurality of successive words out of storage locations in the circular memory;

maintaining a read pointer indicating a read location in the circular memory;

maintaining a write pointer indicating a write location in the circular memory;

comparing the read and write pointers to determine a number of storage locations by which the write pointer leads the read pointer;

reading one of the storage locations twice in response to detecting that the number of storage locations by which the write pointer leads the read pointer is less than a predetermined minimum number; and

overwriting one of the storage locations in response to detecting that the number of storage locations by which the write pointer leads the read pointer is greater than a predetermined maximum number.

2. The method of claim 1 wherein comparing the read and write pointers to determine a number of storage locations by which the write pointer leads the read pointer is not performed each time one of the storage locations is read.

3. The method of claim 2 wherein comparing the read and write pointers to determine a number of storage locations by which the write pointer leads the read pointer comprises reading the value of one of the read and write pointers when the other of the read and write pointers has a value of 0.

4. The method of claim 1 further comprising

examining a plurality of words in the storage locations,

detecting that one of the plurality of words is a fill word, wherein reading one of the storage locations twice comprises reading the storage location containing the fill word twice, and wherein overwriting one of the storage locations comprises examining the plurality of words in the storage locations,

detecting that one of the plurality of words is a fill word and

overwriting the storage location containing the fill word.

5. The method of claim 4 further comprising determining which of the plurality of successive words are fill words and for each of the plurality of successive words providing an indication of whether the word is a fill word.

6. The method of claim 5 wherein providing the indication comprises adding a bit to each word, setting the bit if the word is a fill word and clearing the bit if the word is not a fill word.

7. The method of claim 4 further comprising:

determining whether the number of storage locations by which the write pointer leads the read pointer is

greater than a high-priority threshold number which is greater than the predetermined maximum number; and

if the number of storage locations by which the write pointer leads the read pointer is greater than the predetermined maximum number but less than the high-priority threshold number, deleting one of the fill words with a low priority, and

if the number of storage locations by which the write pointer leads the read pointer is greater than the high-priority threshold number, deleting one of fill words with a high priority.

8. The method of claim 4 further comprising:

determining whether the number of storage locations by which the write pointer leads the read pointer is greater than a high-priority threshold number which is greater than the predetermined maximum number; and

if the number of storage locations by which the write pointer leads the read pointer is greater than the predetermined maximum number but less than the high-priority threshold number, waiting until a series of more than four consecutive fill words is detected and deleting one of the series of fill words, and

if the number of storage locations by which the write pointer leads the read pointer is greater than the high-priority threshold number, waiting until a series of more than two consecutive fill words is detected and deleting one of the series of fill words.

9. The method of claim 1 wherein if the number of storage locations by which the write pointer leads the read pointer is greater than a high error threshold, an error is indicated and if the number of storage locations by which the write pointer leads the read pointer is less than a low error threshold, an error is indicated.

10. The method of claim 1 wherein comparing the read and write pointers to determine a number of storage locations by which the write pointer leads the read pointer comprises determining a value for the number of storage locations by which the write pointer leads the read pointer which is maintained for at least one half of a clock cycle.

11. The method of claim 1 wherein comparing the read and write pointers to determine the number of storage locations by which the write pointer leads the read pointer is delayed for a predetermined period of time after reading or overwriting one of the storage locations in response to detecting the spacing of the read pointer and the write pointer.

12. A method for buffering data in a device configured to be coupled to other devices in a loop network topology, the method comprising:

writing a plurality of successive words into storage locations in a circular memory indicated by a write pointer;

reading the plurality of successive words out of storage locations in the circular memory indicated by a read pointer;

determining a number of storage locations by which the write pointer leads the read pointer;

reading one of the storage locations which contains a fill word two or more consecutive times in response to detecting that the number of storage locations by which the write pointer leads the read pointer is less than a predetermined minimum number; and

overwriting one of the storage locations which contains a fill word one or more times in response to detecting that the number of storage locations by which the write pointer leads the read pointer is greater than a predetermined maximum number.

13. A buffer system comprising:

a circular buffer configured to store data in a plurality of storage locations, wherein a write position is indicated in the buffer by a write pointer and a read position is indicated in the buffer by a read pointer;

write logic configured to write received data to the storage location indicated by the write pointer;

read logic configured to read data from the storage location indicated by the read pointer; and

control logic configured to compare the positions of the read and write pointers and configured to control the read logic to adjust at least one of the read and write pointers in response to the relative positions of the read and write pointers.

14. The buffer system of claim 13 wherein the control logic is configured to adjust the read pointer to cause a fill word to be read twice in response to detecting that the write pointer leads the read pointer by less than a predetermined minimum amount; and wherein the control logic is configured to adjust the write pointer to cause a fill word to be deleted in response to detecting that the write pointer leads the read pointer by more than a predetermined maximum amount.

15. The buffer system of claim 14 wherein the control logic is configured to act with a high priority to skip a fill word in response to detecting that the write pointer leads the read pointer by more than a high-priority threshold amount and to act with a low priority to skip a fill word in response to detecting that the write pointer leads the read pointer by more than the predetermined maximum amount, but less than the high-priority threshold amount.

16. The buffer system of claim 13 further comprising fill word logic configured to add a fill word bit to each word before the word is stored in the circular buffer and to set the fill word bit to indicate whether the word is a fill word.

17. The buffer system of claim 13 wherein the control logic is configured to compare the positions of the read and write pointers by subtracting the value of one of the pointers from the value of the other of the pointers.

18. The buffer system of claim 17 wherein the control logic is configured to compare the positions of the read and write pointers when one of the pointers has a value of 0.

19. The buffer system of claim 14 wherein the control logic is configured to determine an amount by which the write pointer leads the read pointer and to adjust the read pointer in response to the amount only if the amount is maintained for at least half of a clock cycle.

20. The buffer system of claim 13 wherein the control logic is configured to delay comparing the positions of the read and write pointers after adjusting one of the read and write pointers in response to the relative positions of the read and write pointers.

21. A device configured to be installed in a network having a loop topology, the device comprising:

an input port for receiving incoming data from a preceding device in the loop, wherein the incoming data is clocked by an external clock signal;

an output port for transmitting outgoing data to a subsequent device in the loop, wherein the outgoing data is clocked by an internal clock signal;

a circular buffer having a plurality of storage locations, wherein the circular buffer is coupled to the input and output ports and configured to store data in the plurality of storage locations;

write logic coupled to the input port and the circular buffer, wherein the write logic is configured to write received data to storage locations in the circular buffer indicated by a write pointer, wherein the write pointer is advanced after data is written to the indicated storage locations, and wherein data is written to the circular buffer at rate of the external clock signal;

read logic coupled to the output port and the circular buffer, wherein the read logic is configured to read data from the storage locations in the circular buffer indicated by a read pointer and transmit the read data to the output port, wherein the read pointer is advanced after data is read from the indicated storage locations, and wherein data is read from the circular buffer at rate of the internal clock signal; and

control logic configured to determine whether the read logic is reading data out of the circular buffer more quickly or more slowly than the write logic is writing data into the circular buffer, wherein if the read logic is reading data out of the circular buffer more quickly or more slowly than the write logic is writing data into the circular buffer, the control logic is configured to modify the position of at least one of the read and write pointers to compensate therefor.

22. The device of claim 21 wherein if the control logic determines that the read logic is reading data out of the circular buffer more quickly than the write logic is writing data into the circular buffer, the control logic is configured to position the read pointer to cause one of the storage locations in the circular buffer to be read two consecutive times.

23. The device of claim 22 wherein the storage location in the circular buffer which is read twice contains a fill word.

24. The device of claim 21 wherein if the control logic determines that the read logic is reading data out of the circular buffer more slowly than the write logic is writing data into the circular buffer, the control logic is configured to position the write pointer to overwrite one of the storage locations in the circular buffer one time.

25. The device of claim 24 wherein the storage location in the circular buffer which is skipped contains a fill word.

26. The device of claim 21 wherein the control logic is configured to determine whether the read logic is reading data out of the circular buffer more quickly or more slowly than the write logic is writing data into the circular buffer by comparing the positions of the read and write pointers.

27. The device of claim 26 wherein if the write pointer leads the read pointer by more than a predetermined maximum amount, the read logic is reading data out of the circular buffer more slowly than the write logic is writing data into the circular buffer and wherein if the write pointer leads the read pointer by less than a predetermined minimum amount, the read logic is reading data out of the circular buffer more quickly than the write logic is writing data into the circular buffer.